COMPUTACION
Unidad 2

Arquitecturas convencionales

Cátedra de Computación

BIBLIOTECA DEL
CICLO BASICO

2





EUDEBA S.E.M.
Fundada por la Universidad de Buenos Aires

EDITORIAL UNIVERSITARIA DE BUENOS AIRES
Sociedad de Economía Mixta
Rivadavia 1571/73
Hecho el depósito que marca la ley 11.723
ISBN 950-23-0144-7
IMPRESO EN LA ARGENTINA

UNIDAD II

ARQUITECTURAS CONVENCIONALES

OBJETIVOS:

1) Explicar el funcionamiento interno de una computadora tipo Von Newman, a nivel de transferencias entre registrso, mostrando el flujo de señales de control y datos en los ciclos de Fetch y ejecución.

2) Dar noción de programa almacenado, formato de instrucciones y distintos tipos de instrucciones.

3) Mostrar las distintas capas jerárquicas en que puede subdividirse el Hardware, y su interrelación con el Software.

PROGRAMA ANALITICO:

- Arquitectura de computadoras: subsistemas componentes de una computadora. Memoria; CPU; Entrada/Salida.
- Ciclos de operación: Fetch y Ejecución. Concepto de instrucción; formatos Flujo de datos entre registros. Busses. División del Hardware en capas jerárquicas (U.C., Registros y Acu, Memoria, Interfases y Periféricos)

ARQUITECTURA DE COMPUTADORAS DIGITALES

Para introducir las definiciones y conceptos básicos sobre principios de funcionamiento y arquitectura de máquinas timo Von Neumann, se utilizará un modelo de las mismas realizado con elementos de la vida diaria, para su mejor comprensión.

A tal fin, se tomará un casillero como el existente en los hoteles para guardar las llaves y la correspondencia pertenecientes a cada habitación; este elemento cumplirá las funciones de lo que más adelante será definido como "Memoria Central" (ver figura 1). Obsérvese que cada casilla está numerada; es decir tiene asociado un número correspondiente a la habitación a la cual corresponde, y que sirve para identificarla. Así por ejemplo , el casillero de la figura 1 tendrá casillas numeradas del 0 al 500. A través de estos números, cualquier persona (en un hotel, el conserje por ejemplo) puede "dirigirse" a una casilla en particular, o sea que estos números cumplen la función de Dirección de cada casilla.

Por otro lado cada casilla puede "contener" en su interior cartas, papeles, etc. que el empleado colocará y sacará de acuerdo a las necesidades. En el ejemplo que se está considerando, sólo se podrá guardar en cada casilla o celda una sola tarjeta en la que se pueda escribir un número de N dígitos como máximo. Es evidente entonces que cada celda tendrá asociada dos parámetros: uno es el número que identifica la celda que fué llamado Dirección de la casilla, y que no se puede cambiar, y el otro número es el que está escrito en la tarjeta que está guardada dentro de la celda, que por analogía se llama Contenido de la celda, y puede ser alterado cuantas veces se desee.

Dirección y Contenido son, pues, dos características de cada celda de memoria, siendo la Dirección la que permite acceder al Contenido. Al número escrito en la tarjeta se lo denomina "Palabra" de Memoria, y a la cantidad máxima de dígitos que puede contener esta palabra (N), se la denomina "Longitud" de la palabra, y en cierta forma caracteriza la longitud de palabra de toda la máquina.

Estos contenidos de cada celda pueden ser "leídos" o "escritos"; es decir: se puede extraer una tarjeta de una determinada celda especificada por su dirección,"leer" el número escrito en ella, y luego guardarla en la misma casilla, o, por el contrario, al extraer la tarjeta borrar su contenido, "escribir" otro número en la misma y luego guardarla en la celda de donde fue tomada. Estos son los llamados "ciclos" de lectura o escritura.

Otro elemento a usar en este modelo trivial de computadora, es una máquina de calcular, en la que la computadora realizará las operaciones aritméticas, y por lo tanto hace las veces de "Unidad Aritmética" de la computadora. Esta calculadora tendrá un teclado por el que se introducen los números y operaciones, y un visor en donde aparecen los operandos y los resultados. El visor permite guardar o almacenar temporariamente al operando o resultado hasta que es reemplazado por otro. Es por lo tanto un lugar donde se puede guardar temporariamente un número, y a todo elemento que permita hacer esto (es decir: guardar temporariamente un número), se lo denomina "Registro". En particular, a este registro de la Unidad Aritmética se lo llama "Acumulador". Se verá más adelante, que la computadora tiene una serie de estos registros, dedicados a funciones específicas.

Se usará en este modelo una máquina de escribir para simular la "unidad de Salida", es decir el elemento que usará la computadora para imprimir los datos que desee transmitir hacia el exterior. Un objetivo análogo tendrá un tarjetero que contendrá tarjetas que serán cargadas desde el exterior, conteniendo datos que serán procesados; en este caso este elemento cumplirá funciones de "Unidad de Entrada".

Se deberá proveer además un reloj cuenta ganado, es decir un contador mecánico con 2 perillas; una de ellas al ser pulsada, aumenta en uno la cuenta (el número) existente en el contador, y con la otra se puede "escribir" cualquier número en el contador. Obsérvese que de acuerdo a lo que se definió anteriormente, este contador puede ser considerado también como un Registro. Junto con este reloj, deberán adicionarse algunas tarjetas en blanco, semejantes a las usadas en los casilleros de memoria y cuyas funciones será explicadas más adelante.

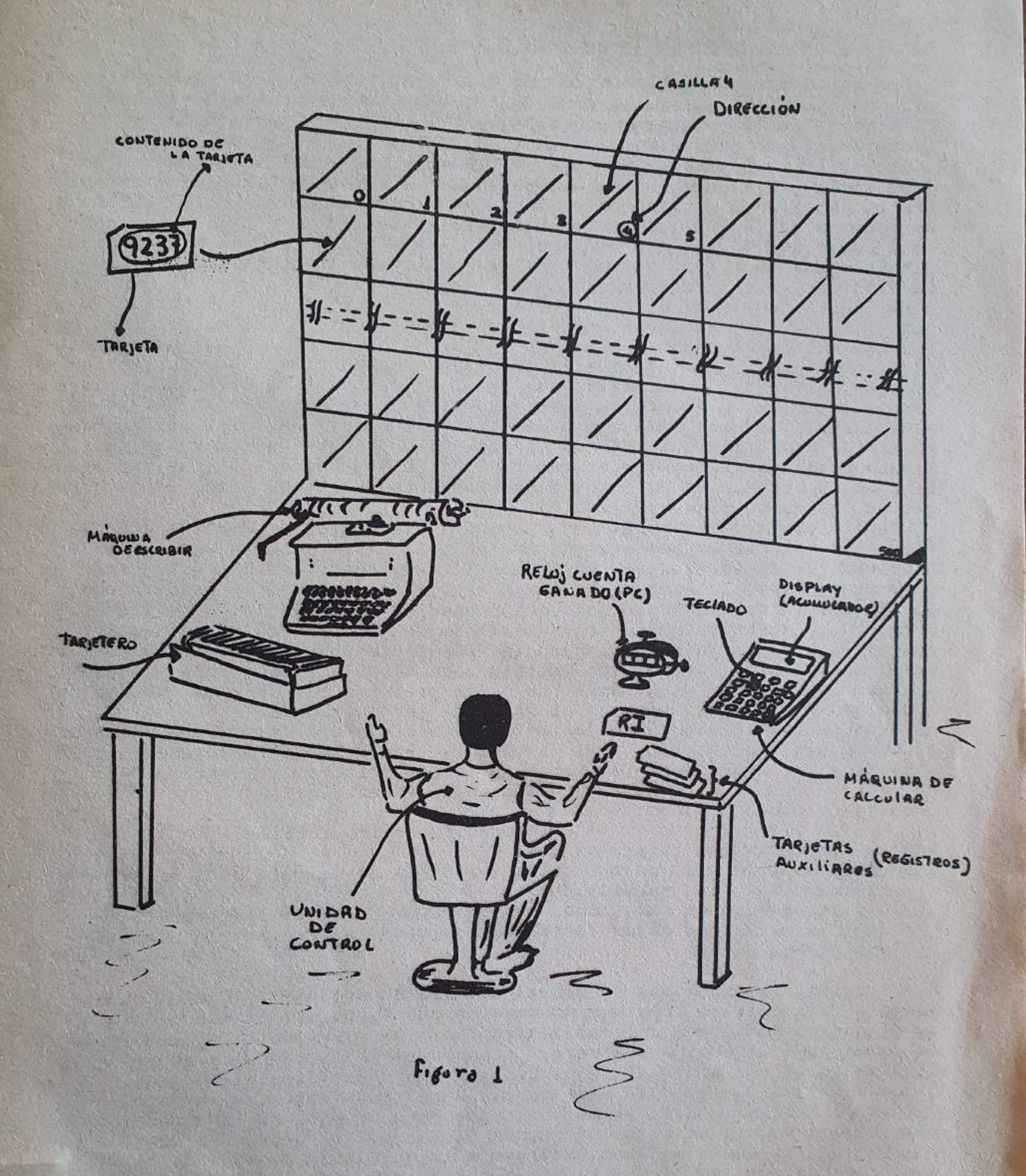

CASILLA 4
DIRECCIÓN
CONTENIDO DE LA TARJETA
9237
TARJETA
0
1
2
3
4
5
MÁQUINA DE ESCRIBIR
RELOJ CUENTA GANADO (PC)
DISPLAY (ACUMULADOR)
TECLADO
TARJETERO
RI
MÁQUINA DE CALCULAR
TARJETAS AUXILIARES (REGISTROS)
UNIDAD DE CONTROL

Figura 1

Todos estos elementos descriptos, simulan las distintas partes constitutivas de una computadora, pero por si solos no funcionan como tal; falta una elemento más que los maneje y los haga funcionar. Para este fin se incorporará al modelo, a una persona especialmente entrenada. Esta persona cumplirá funciones de "Unidad de Control". Esta persona está entrenada para "ejecutar" órdenes. Estas órdenes o "instrucciones" estarán escritas en algunas de las tarjetas guardadas en las casillas de la Memoria.

La Unidad de Control tomará de allí las instrucciones que tiene que ejecutar. Esto implica que -como se verá más adelante- todo proceso que deba realizar la computadora a través de las unidades descriptas anteriormente, deberá ser descompuesto (o subdividido) en una serie (o secuencia) de órdenes (o instrucciones), que al ser ejecutadas una tras otra, permitan realizar el proceso deseado. Esto implica que la Unidad de Control deberá tomar la primer instrucción y ejecutarla, luego la segunda, luego la tercera y asi sucesivamente hasta llegar a la última, con lo que habrá concluido de ejecutar el proceso deseado. Esta secuencia de instrucciones es lo que más adelante se definirá como "Programa". Como se dijo, estas instrucciones estarán escritas en las tarjetas contenidas en el casillero de memoria, pero como alli sólo se puede escribir números, estas deberán escribirse codificadas, es decir si se quiere escribir una orden (instrucción) para sumar, se deberá escribir un número que la Unidad de Control interpretará como una orden (o instrucción) de suma. Por lo tanto se le deberá dar a la Unidad de Control una lista de todas las instrucciones posibles, junto con el número (código) con el que se las representará en la tarjeta. Como se verá más adelante, existen varios tipos distintos de instrucciones.

Para realizar (o ejecutar) un programa, la Unidad de Control deberá "leer" la tarjeta de memoria que contenga la primer instrucción, copiar su contenido en una tarjeta en blanco llamada "Registro de Instrucción", guardar la tarjeta leída en el mismo casillero de donde la sacó, luego analizar e interpretar la instrucción escrita en el Registro de Instrucción , y luego ejecutarla. Terminada la ejecución deberá "ir a buscar" y leer la tarjeta de memoria que contiene la segunda instrucción, copiarla en el Registro de Instrucción (borrando de éste la instrucción anterior), guardar la tarjeta leida, analizar, interpretar y ejecutar la segunda instrucción, y así siguiendo con el resto de instrucciones.

Como se ve, la Unidad de Control repite siempre el mismo procedimiento 1) busca una instrucción, y 2) la ejecuta, luego busca la siguiente y la ejecuta y asi sucesivamente. Estos se llaman "Ciclos de Instrucción"

Ciclo de Instrucción { Búsqueda de instrucción
Ejecución de instrucción.

Acá se presentan dos interrogantes:

1.- ¿Cómo sabe la U.C. en que celda (casilla) de memoria está la primera instrucción y las subsiguientes?
2.- Las instrucciones para poder ser ejecutadas (por ej.:instrucción de sumar), necesitan datos (operandos); ¿Dónde se encuentran almacenados dichos datos?

El primer interrogante se resuelve a través del uso del reloj cuenta ganado. Para ello se toma como convención que la primer instrucción que se ejecutará habrá que colocarla siempre en la misma celda, por ejemplo la celda cuya dirección sea cero. Entonces al iniciarse el trabajo de la máquina habrá que poner al reloj cuenta ganado en forma automática, en la dirección elegida (en este caso cero). Habrá que indicarle entonces a la Unidad de Control que la primer instrucción la busque en la dirección de memoria que figura en el reloj cuenta ganado.¿Y las restantes instrucciones, dónde las busca?. Si se instruye a la Unidad de Control de tal manera que después que lea la tarjeta correspondiente a una instrucción, la copie y la vuelva a guardar, pulse la palanca que incrementa en uno la cuenta del reloj cuenta ganado, allí tendrá la dirección de la próxima instrucción. Para que este método funcione, se deberán colocar las instrucciones en celdas contiguas de memoria, este reloj cuenta ganado se ha transformado entonces en el registro que contiene la dirección de la próxima instrucción a ser ejecutada y se lo llama por lo tanto "Registro Contador de Programa (P.C.)"

La secuencia de pasos que deberá realizar la Unidad de Control para completar un ciclo de instrucción será, pues, la siguiente:

1.- Búsqueda de la instrucción en la dirección que indica el contador de progama.
2.- Lectura de la misma y copia en el registro de instrucción.
3.- Guardar la tarjeta en la celda de donde la sacó.
4.- Incrementar el contador de programa.
5.- Analizar e interpretar la instrucción.
6.- Ejecutar la instrucción.

El segundo interrogante se refería al lugar de donde deberían obtenerse los datos necesarios para ejecutar la instrucción: es evidente que habiendo una memoria donde se pueden almacenar números (y que hasta ahora se ha usado solamente para guardar instrucciones en forma codificada), sea esta memoria la que se use para guardar también los datos. Pero entonces ¿cómo y dónde se especifica la dirección de las celdas de memoria que contienen los datos necesarios para la ejecución de cada instrucción? Normalmente esta dirección está contenida (en forma directa o indirecta) en la propia instrucción. Esto es: la tarjeta que contiene la instrucción en forma codificada, contendrá además la o las direcciones del o de los operandos. Así, el formato típico de una instrucción podrá ser dibujado de la siguiente manera:

C.Op.	D1	D2	D3

Siendo:
C.Op. el código que identifica a la instrucción, llamado Código de Operación.
D1 la dirección del primer operando
D2 la dirección del segundo operando
D3 la dirección donde se guarda el resultado.

Este tipo de formato de instrucción corresponde a máquina llamada de tres direcciones, en donde la longitud de palabra de la misma es grande. En máquinas donde la longitud de palabra no es tan grande (minicomputadoras, microcomputadoras) se usa formatos de dos direcciones y aún hasta de una dirección.

Formato de 2 direcciones:

C.Op.	D1	D2

Siendo:
D1 la dirección del primer operando y del resultado
D2 la dirección del segundo operando

Formato de una dirección:

C.Op.	D1

Siendo:
D1 la dirección del primer operando.

El otro operando (por ej. en el caso de una suma) se considera que ya está almacenado en el acumulador, registro en el que también queda guardado el resultado.

CLASIFICACION DE LAS INSTRUCCIONES

Es casi inmediato pensar que deberán existir instrucciones de tipo aritmético, como por ejemplo sumar, restar,comparar, etc. o instrucciones que realicen operaciones de tipo lógico como por ejemplo Y , O , XOR, etc. No es tan evidente considerar la existencia de otro tipo de instrucciones como por ejemplo las de control:cuando se describió la necesidad de la existencia del contador de programa se dijo que para que el método funcionara era necesario colocar en secuencia las instrucciones en memoria. Pero esto no siempre es posible. Basta pensar en un programa que al finalizar tenga que volver a empezar, para entender que la secuencia debe ser rota en alguna parte. Otro tipo de ruptura de secuencia es la condicional, o sea que la secuencia se altera o no dependiendo de una condición. Así, por ejemplo, en ùn programa que resuelva una ecuación de 2°grado, si el radicando es positivo, negativo o cero, el camino a seguir para resolver cada caso será distinto y por lo tanto habrá que "saltar" a una de tres secuencias que por supuesto estarán ubicadas en lugares distintos de la memoria.

Estas consideraciones hacen necesaria la existencia de dos tipos de instrucciones de control: instrucciones de salto incondicional e instrucciones de salto condicional (en función estas últimas de ciertos indicadores o "flags" que entrega la Unidad Aritmética).

Otro grupo de instrucciones necesarias para el manejo de las Unidades de Entrada/Salida son precisamente las instrucciones de Entrada/Salida.

Se dará a continuación un listado de un posible conjunto de instrucciones (sin hacer referencia a ninguna máquina en particular), agregando una codificación totalmente arbitraria, y por razones didácticas en decimal (por supuesto en una máquina real estos códigos de operación estarán expresados en binario). El formato es de una dirección.

INSTRUCCIONES ARITMETICO - LOGICAS

Código de Operación	Mnemónico	Descripción
10	SUM,D	Suma el contenido del Acumulador al connido de la direccion D.
11	RES,D	Resta el contenido del Acumulador al contenido de la dirección D.
12	COMP,D	Compara el contenido del Acumulador, con el contenido de D.
13	INV	Complementa el acumulador.
14	CAR Ac,D	Carga el Acumulador con el contenido de D.
15	MOV Ac,D	Copia el contenido del Acumulador en la dirección D.
16	Y,D	Y, del Acumulador con el contenido de D.
17	O,D	O el Acumulador con el contenido de D.
18	XOR,D	Xor, del Acumulador con el contenido de D.
19	CORR I	Desplaza 1 bit a la izquierda el Acumulador.
20	CORR D	Desplaza 1 bit a la derecha el Acumulador.
21	BORR Ac	Pone ceros en el Acumulador

INSTRUCCIONES DE CONTROL

30	SALTAR,D	La próxima instrucción la obtienen en la dirección D
31	SALTARO,D	Si la última operación aritmética dió cero, salta a D; de lo contrario no salta.
32	SALTARN,D	Idem anterior, pero para resultado negativo.
33	SALTARP,D	Idem anterior, pero para resultado positivo.
34	SALTAROV,D	Idem anterior, pero por 'over flow'.
35	PARAR	Para el funcionamiento de la máquina.

INSTRUCCIONES DE ENTRADA/SALIDA

40	ENTRAR	Carga en el Acumulador el dato que está en la unidad de entrada de dirección D
41	SALIR,D	Transfiere el contenido del Acumulador a la Unidad de salida de dirección D.

Se verán a continuación los pasos a seguir por el modelo explicado anteriormente,(considerando que cuenta con el conjunto de instrucciones antes descripto), para realizar la suma del número que está en la dirección D1 al número que está en la dirección D2, dejando el resultado en D3. (en el ejemplo se toma D1=50;D2=51 y D3=52)

El programa a ejecutar será el siguiente:

DIRECCION	CONTENIDO	
0	14 | 50	CARGA AC. CON EL CONTENIDO DE D1
1	10 | 51	SUMA D2 AL AC.
2	15 | 52	COPIA EL CONTENIDO DEL AC EN D3

Obsérvese que este mismo proceso de suma, en una máquina de tres direcciones se hace con una sola instrucción.

Los pasos a seguir por el modelo descripto anteriormente para ejecutar este programa son los siguientes:

1.- PC=0 (dirección de arranque)

2.- LEE CONTENIDO DIRECCION 0 --→14 50 (tarjeta de celda 0)

3.- RI=1450 (RI--→registro de instrucción)

4.- GUARDA TARJETA QUE LEYO EN CELDA 0

5.- PC←--PC+1 (PC=1)

6.- ANALIZA RI (instrucción de CARGAR desde D1=50)

7.-LEE CONTENIDO DIRECCION 50 --→27 (primer operando)

8.- DIGITA 27 EN LA MAQUINA DE CALCULAR CON ENTER

9.- GUARDA TARJETA QUE LEYO EN CELDA 50

FIN EJECUCION PRIMERA INSTRUCCION

10.- LEE CONTENIDO DIRECCION 1 (PC=1)--→10 51

11.- RI=1051

12.- GUARDA TARJETA QUE LEYO EN CELDA 1

13.- PC ←--PC+1 (PC=2)

14.- ANALIZA RI (instruccion de SUMAR AC.+ contenido D2)

15.- LEE CONTENIDO DIRECCION 51 --→12 (segundo operando)

16.- DIGITA 12 ENLA MAQUINA DE CALCULAR CON TECLA +.(como resultado Ac.=27+12=39)

17.- GUARDA TARJETA QUE LEYO EN CELDA 51

FIN EJECUCION SEGUNDA INSTRUCCION

18.- LEE CONTENIDO DIRECCION 2 (PC=2) --→15 52

19.- RI=1552

20.- GUARDA TARJETA QUE LEYO EN CELDA 2

21.- PC ←-- PC+1 (PC=3)

22.- ANALIZA RI (instrucción de MOVER el Ac. a la dirección D3=52)

23.- BUSCA LA TARJETA CONTENIDA EN LA DIRECCION 52, LA BORRA Y COPIA EL Ac. = 39

24.- GUARDA LA TARJETA QUE EXTRAJO DE LA CELDA DE DIRECCION 52 EN SU LUGAR.

FIN TERCERA INSTRUCCION.

Como se puede ver, la ejecución de un programa consiste en la transferencia de datos entre los distintos registros, la memoria y eventualmente las unidades de entrada/salida, pasando (o no) por la unidad aritmética.

A modo de ejemplo se describirá un programa que sume los elementos de un vector contenidos en las direcciones 60 a 69, dejando el resultado de la suma en 70

DIRECCION	CONTENIDO	
00	21	00
01	10	60
02	15	70
03	14	01
04	12	48
05	31	09
06	10	49
07	15	01

08	30	01
09	35	00
.		
.		
.		
.		
48	10	69
49	00	01

En el ejemplo anterior se han usado instrucciones de salto condicional e incondicional.

¿Cómo hace una Unidad de Control del modelo que se está describiendo para ejecutar una instrucción de salto? Al analizar el Registro de Instrucción e interpretar que se trata de un salto (condicional o incondicional), toma la dirección que figura junto al código de operación y, si se trata de un salto incondicional, modifica el número existente en el reloj cuenta ganado (contador de programa) usando la palanca correspondiente, reemplazándolo por la dirección antes mencionada. (si fuese un salto condicional, antes de modificar al PC, debe verificar que se cumpla la condición especificada).

FLUJO DE DATOS Y DE CONTROL: BUSES

Tomando como referencia el modelo descripto , se observa que el funcionamiento de la computadora se limita a transferir datos entre registro, memoria, Acumulador y periféricos. En la figura 2 se ve un diagrama en bloques de una CPU real; se observan los registros ya mencionados , a los que se agregan algunos otros de uso general.

La Unidad de Control se limita a abrir y cerrar compuertas de tal manera de lograr transferencias de datos entre los distintos registros, siguiendo los esquemas ya descriptos.

Entre los registros que aparecen en la figura 2, se encuentran MDR y MAR. MDR (Registro de Dato de Memoria) es el registro que se usa para transferir los datos entre la memoria y la CPU; y MAR (registro de direcciones de memoria) es el registro que se usa para "cargar" las direcciones de las celdas sobre las que se transfieren los datos del registro MDR. Por ejemplo, si se quiere leer el contenido de la celda 123, se coloca el 123 en MAR, se da orden de lectura, y al cabo de un tiemplo aparecerá en MDR el contenido de la dirección 123. Para escribir por ejemplo 203 en la dirección 54 habrá que colocar 203 en el MDR y 54 en el MAR.

La Unidad de Control deberá generar entonces una secuencia de señales que abran y cierren "compuertas" (señales de control) para encaminar el flujo de datos a través de la máquina.

A fin de clarificar el funcionamiento del diagrama en bloques de la figura 2, se listará la secuencia de señales que deberá generar la CPU para ejecutar SUM,D (se supone que el otro operando esta ya cargado en el Ac.)

1.- PC_{sale}, MAR_{entra}, LEER --→ Copia el contenido del PC en el MAR, y da lectura de memoria.

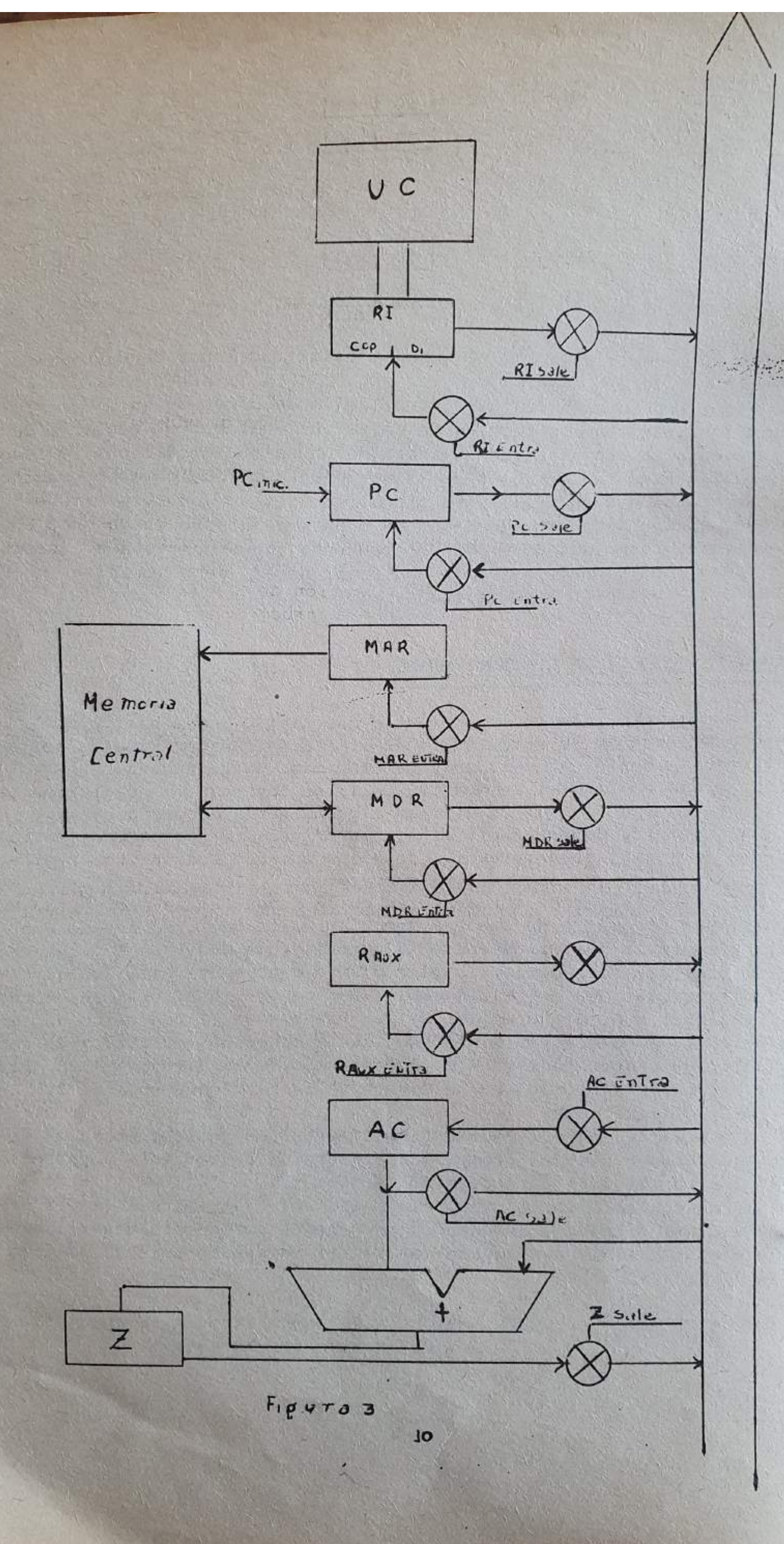

UC
RI
CCP
Di
RI sale
RI entra
PC inc.
PC
PC sale
PC entra
MAR
Memoria
Central
MAR entra
MDR
MDR sale
MDR entra
RAUX
RAUX entra
AC entra
AC
AC sale
+
Z sale
Z

Figura 3

2.- MDR_{sale}, RI_{entra}, PC_{inc} --→ Copia el contenido del MDR (que es el registro en el que la memoria puso el contede la dirección cargada anteriormente en el MAR) el el RI, e incrementa el PC

FIN DEL CICLO DE BÚSQUEDA: LA INSTRUCCIÓN SUM,D ESTA EN RI

3.- CAMPO DE DIRECCIÓN DE RI_{sale}, MAR_{entra}, LEER --→ Copia la dirección D1 del dato a sumar en MAR, y lee la memoria.

4.- MDR_{sale}, SUM, Z_{entra} --→ Copia el contenido de MDR (Dato que estaba en dirección D1) en el SUMADOR, suma, copia el resultado en el registro auxiliar Z.

5.- Z_{sale}, $Ac._{entra}$ --→ Copia el resultado de la suma que quedó temporariamente guardado en Z al acumulador

6.- FIN --→ Indica que terminó la ejecución de SUM,D1, lo que habilita la búsqueda de la próxima instrucción.

A esta secuencia de operaciones que debe hacer la Unidad de Control para ejecutar la instrucción Dum D, se la llama MICROPROGRAMA; esto mismo ocurre con cada instrucción. Se ve entonces, que para ejecutar cada instrucción de programa, la Unidad de Control deberá ejecutar un MIcroprograma. Este es el nivel más bajo de programación que existe, y estos micriprogramas generalmente estan almacenados en una memoria no volátil (que no se borra), obteniendo la Unidad de Control el microprograma correspondiente a cada instrucción a partir del código de operación de la misma. (Es decir, usa el código de operación para obtener la dirección de arranque del microprograma que ejecuta a dicha instrucción).

Esta Unidad de Control antes descripta es la llamada Unidad de Control MIcroprogramada.Es conveniente aclarar que no es la única posibilidad que existe para construir una Unidad de Control.

DIVISION EN CAPAS JERARQUICAS

Teniendo presente lo descripto anteriormente, se puede subdividir al HARDWARE (y posteriormente al SOFTWARE) en capas jerárquicas, a la manera de las capas que conforman una cebolla. Esta subdivisión se deberá hacer de tal manera que cada capa esté apoyada o sustentada en las capas que le son internas, y sea independiente y le sirva de apoyo a sus capas externas . (Ver figura 3). De acuerdo a este razonamiento, el núcleo o zona más interna en esta subdivisión, está constituido por la Unidad de Control y su microprograma. Sobre ella se apoyan los registros,acumulador, los que a su vez sirven de sustento a la Memoria Central. Sobre toda esta estructura, cuya capa externa es la Memoria (constituyendo hasta aquí el procesador Central), se construyen las interfases de Entrada/salida, que a su vez manejan a los periféricos respectivos.

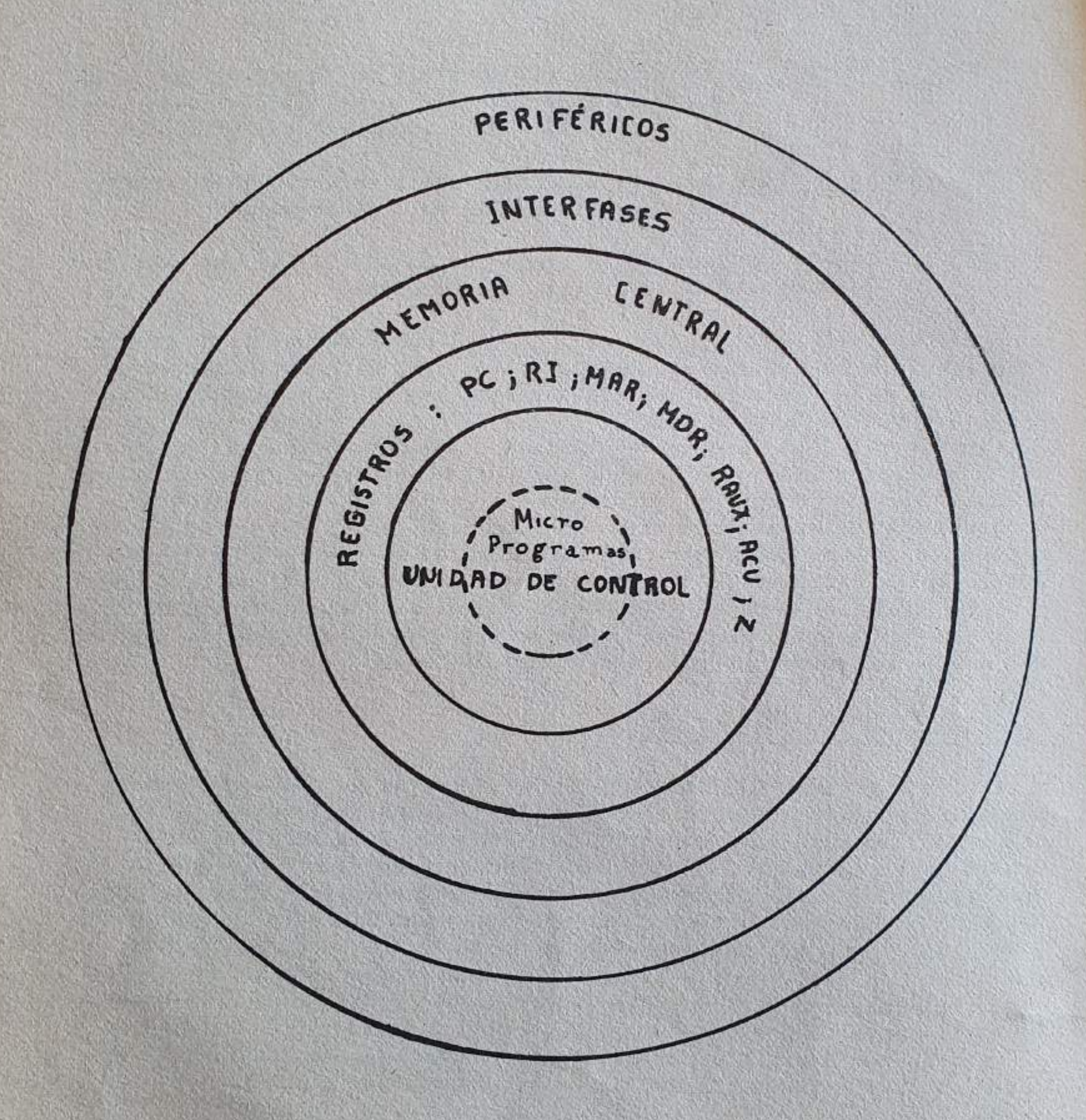

PERIFÉRICOS
INTERFASES
MEMORIA CENTRAL
REGISTROS : PC ; RI ; MAR; MDR; RAUX; ACU ; Z
Micro
Programas
UNIDAD DE CONTROL

Figura 3